The following publication W. Lin, H. Lan and J. Cao, "Graph Privacy Funnel: A Variational Approach for Privacy-Preserving Representation Learning on Graphs," in IEEE Transactions on Dependable and Secure Computing, vol. 22, no. 2, pp. 967-978, March-April 2025 is available at https://doi.org/10.1109/TDSC.2024.3417513.

Graph Privacy Funnel: A Variational Approach for Privacy-Preserving Representation Learning on Graphs

Wanyu Lin, *Member, IEEE,* Hao Lan, *Member, IEEE,* Jiannong Cao, *Fellow, IEEE*

Abstract—This paper investigates the problem of learning privacy-preserving graph representations with graph neural networks (GNNs). Different from existing works based on adversarial training, we introduce a variational approach, called *vGPF*, to encourage the isolation of sensitive attributes from the learned representations. Specifically, we first formulate a non-asymptotic information-theoretic problem for characterizing the best achievable privacy subject to the utility constraints of graph representations, termed as **G**raph **P**rivacy **F**unnel (GPF). Then we theoretically analyze that the GPF objective can be directly optimized over through a variational approximation upper bound. *vGPF* allows us to parameterize the privacy-preserving graph mapping with GNN encoders and use the reparameterization trick for training. Compared with existing adversarial approaches, *vGPF* exhibits more stable predictive performance as it does not rely on an additional adversarial network that may incur training stability in practice. Experiments across multiple datasets from various domains demonstrate that *vGPF* outperforms its state-of-the-art alternatives in terms of predictive accuracy, performance stability, and robustness to attribute inference attacks. We also show that *vGPF* enjoys high flexibility in the sense that it is compatible with various graph learning tasks with different GNN encoder architectures, and it can enforce privacy over any combinations of sensitive attributes in one shot.

Index Terms—Privacy-Preserving Graph Representation Learning, Graph Neural Networks, Information Funnel, Variational Approach.

1 Introduction

Graph neural networks (GNNs) are deep learning models tailored to learn and model information structured as graph data. They have demonstrated impressive performance for graph representation learning across a wide range of disciplines, such as recommender systems [1], financial systems [2], and social information systems [3]. Nevertheless, previous studies have shown that adversary with the access to the graph representations, can infer sensitive attributes of interests such as gender and race [4], [5], coined as attribute inference attacks [4], [5], [6]. Yet, the neighborhood aggregation scheme of GNNs exposes additional vulnerabilities to adversaries seeking to extract sensitive information [7]. The privacy vulnerabilities of GNNs significantly hinder the applicability of these models on privacy-sensitive applications [8]. For example, in recommender systems, we may want to boost the recommendation performance by incorporating social relationships in social graphs, but we do not want to expose specific sensitive attributes of users (e.g., age or gender) through learned graph representations.

There exist a few works attempting to generate graph representations that can defend against attribute inference attacks [9], [10]. However, these prior works are based on adversarial training. Their estimation quality heavily relies on the adversarial network's modeling performance or training stability, leading to a sub-optimal strategy against attribute inference attacks. To this end, we propose a variational approach that does not rely on an additional adversarial network for training. In particular, our proposed approach falls into the variational auto-encoding architecture (i.e., VGAE [11]) consisting of two neural networks—a GNN encoder for extracting the graph representations and a decoder for the original prediction task, and it is trained with the *privacy funnel principle*—an information-theoretic principle initially coined by Makhdoumi *et at.* [12]. In essence, the privacy funnel (PF) problem is proposed to encode a dataset into privacy-preserving representations, where the informativeness of representations is measured by mutual information. The PF principle is critical for evaluating data utility and privacy leakage from an information-theoretic perspective. This principle naturally leads us to optimize a non-asymptotic information-theoretic formulation for characterizing the best achievable privacy subject to the utility constraints of graph representations. We term our proposed optimization problem as **G**raph **P**rivacy **F**unnel (GPF). If solving the GPF problem is feasible, we can quantify the trade-off between utility and privacy of graph representation learning.

However, extending the PF principle to privacy-preserving/private representation learning on graph-structured data presents several unique challenges. First, prior works on the PF principle assume that the training samples in the dataset are independent and identically distributed (i.i.d.), such as tabular data and image [12], [13]. This assumption does not hold for graph-structured data, making the model training under the PF principle hard. Second, due to the non-Euclidean property of graph-

- *Wanyu Lin is the corresponding author and is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: wanylin@comp.polyu.edu.hk.*
- *Hao Lan is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: lanhao@mail.tsinghua.edu.cn.*
- *Jiannong Cao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: jiannong.cao@polyu.edu.hk.*

structured data and the intractability of mutual information and conditional mutual information [1], the privacy funnel problem is hard to optimize over. To this end, we first adopt the local-dependence assumption of graph-structured data to overcome the challenge induced by non-i.i.d. property. This assumption has been widely used for constraining the search space of optimal representation learning within a graph [14]. As the GPF objective includes mutual information that is intractable, we propose to use variational inference and construct two variational bounds leading to a final objective that is tractable for optimization. We call the resulting approach variational graph privacy funnel (*vGPF*). This variational approach parameterizes the privacy-preserving graph representation learning with GNN encoders and uses the reparameterization trick for training.

Highlights of our original contributions are as follows. We formulate a GPF problem to characterize the best achievable privacy subject to the utility constraints of graph representations. To the best of our knowledge, while the notion of privacy funnel has been used for privacy-preserving mapping on the i.i.d. dataset (e.g., tabular data and images) [12], this is the first effort from the PF principle to learn privacy-preserving representations on graph-structured data with GNNs. We theoretically analyze and quantify the privacy-utility trade-off by introducing tractable variational bounds leading to the GPF objective. Different from prior works based on adversarial training [9], [10], the GPF is a single information-theoretic objective that can be directly optimized over. Our framework is a variational approach and does not rely on an additional adversarial network that may incur training stability issues in practice. Moreover, by systematically evaluating the robustness of learned representations against various types of adversaries, including a worst-case adversary, we demonstrate that *vGPF* can learn graph representations with better predictive performance and stability yet higher privacy guarantee than state-of-the-art methods. We also empirically show that *vGPF* exhibits high flexibility for various graph learning tasks with different GNN encoder architectures, and it can remove multiple sensitive attributes in one shot yet does not require modifying the model design.

2 Problem Setup

Notations. Consider an input graph $\mathbf{G} = (V, E, \mathbf{X})$ with n nodes, where V is the node set, E denotes the edge set, and $\mathbf{X} \in \mathbb{R}^{n \times f}$ are node attributes/features. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ denote the adjacency matrix of $\mathbf{G}$, *i.e.*, $A_{ij} = 1$ if $(i, j) \in E$ or 0 otherwise. For simplicity, the input graph can be represented as $\mathcal{D} = (\mathbf{A}, \mathbf{X})$.

System Model. Consistent with prior works [9], [10], we focus on relation/link prediction tasks based on GNNs. The objective is to obfuscate sensitive information from the learned representations based on GNNs. We particularly focus on addressing attribute inference attacks in the inference stage of model partitioning settings [6], [9], [10], [15]. Specifically, a GNN-based graph learning model F is trained to predict target y (e.g., ratings in recommender systems) at a trusted server. Once trained, the entire inference model will include two portions, i.e., a GNN encoder Ψ owned by the data owner and a prediction network Φ owned by a client. This reflects the real-world scenario in the context of Artificial Intelligence of Things (AIoT) [15], where a machine learning model is distributedly deployed across various mobile devices, either due to the data privacy concern or resource constraints.

Threat Model. We assume that two parties, i.e., the data owner and the client, communicate over a noiseless channel in the inference stage (see Fig. 1(a)). The data owner is to map the input graph $\mathcal{D}$ into low-dimensional vectors $\mathbf{Z}$ with the GNN encoder, i.e., $\mathbf{Z} = \Psi(\mathcal{D}) \in \mathbb{R}^{n \times f'}$, while the client can access the representations from the data owner via query. The client does not access the GNN encoder or know its parameters, but she can use the representations for further analysis with the prediction network, i.e., $Y = \Phi(\mathbf{Z})$. She is also an adversary who has access to an auxiliary dataset including node index and corresponding attributes, e.g., $(V, \mathbf{S})$ for training an attack model to infer sensitive attributes $\mathbf{s}$ based on the received representations. We assume that the client is passive but computationally unbounded. Without loss of generality, we consider a setting that the sensitive variables $\mathbf{s}$ can not be directly accessible based on $\mathcal{D}$, i.e., $\mathbf{s} \not\subset \mathbf{x}$, but it can be inferred from $\mathcal{D}$ due to "overlearning." The "overlearning" phenomenon addressed that a model trained for a seemingly simple objective unintentionally learned privacy- and bias-sensitive attributes, even though these sensitive variables are not in the training data [6].

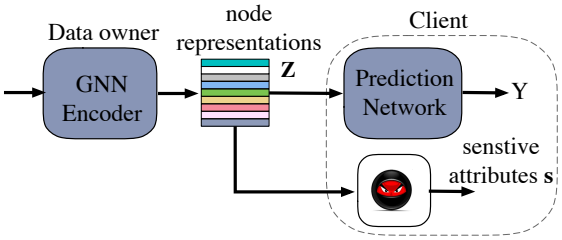
Objective. Our ultimate goal is to obtain a GNN encoder for generating graph representations that can defend against attribute inference attacks in the inference stage of model partitioning settings, while achieving the best possible predictive performance on the target task, i.e., relation prediction tasks.

3 Related Work

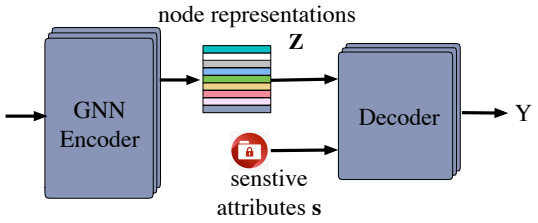
GNN Encoder. A GNN encoder incorporates graph structure and node features into low-dimensional node representations, which are informative and can be used for various downstream tasks, such as node classification and link prediction. With graph pooling, these node representations can be fused as graph representations, thus can be used for graph-level tasks, such as graph classification and graph regression. For a GNN encoder with L layers, a node's representation can capture the structural information within its L-hop graph neighborhood [16]. There are many types of GNN encoder architectures, such as ChebNet [17], GCN [18], GAT [19], GraphSage [16], and many more. Interested readers may refer to comprehensive surveys on GNNs (e.g., [20]).

Privacy Leakage on GNNs. Previous studies have shown that deep learning models are vulnerable to privacy attacks [4]. The neighborhood aggregation scheme of GNNs exposes additional vulnerabilities [7] to adversaries seeking to extract sensitive information. There exists some prior work on the privacy implication of training GNNs on graphs [5], [7], [21]. Duddu *et al.* analyzed the privacy risks of graph embedding with various attacks, including membership inference attacks, graph reconstruction attacks, and attribute inference attacks. Later, Olatunji *et al.* proposed membership

[1]For simplicity, we do not particularly distinguish "mutual information" and "conditional mutual information" if there is no risk of confusion in this paper.



(a) Inference Stage



(b) Training with GPF

Fig. 1. (a) An illustration of the threat model in the inference stage. The client may want to infer sensitive attributes based on the representations received from the data owner. (b) An illustration of *vGPF* for training privacy-preserving GNN encoder at a trusted server. The decoder admits the representations from the GNN encoder and the sensitive attributes of interests. Once trained, the GNN encoder can extract graph representations that defend against attribute inference attacks in the inference stage.

inference attacks under restricted scenarios, by 0-hop query and 2-hop query to the victim models. He *et al.* showed that given various background knowledge, the outputs of a GNN indeed could be utilized to infer the rich information about the graph structure used to train the model. They coined the attack "link stealing attack." This paper considers the problem of privacy-preserving graph representation learning with GNNs that can defense against attribute inference attacks.

Privacy-Preserving GNNs. A few attempts were proposed to address privacy concerns on graph learning based on GNNs. LPGNN [22] was proposed to preserve feature privacy of nodes with differential privacy, while DPGGAN [23] was for generating synthetic graphs attempting to preserve properties of graph structure with edge differential privacy. Solitude [24] attempted to protect the relation privacy of the graph based on local edge-differential privacy. However, the randomized mechanisms were typically manipulated over the input node features/edges in the training data. PPNE [25] was a privacy-preserving network embedding framework designed to identify the optimal perturbation solution with the best privacy-utility trade-off in an iterative way. Unlike above works, we focus on obfuscating sensitive attributes that may not be explicitly present in the training data but could be inferred based on the learned representations due to "overlearning" [6].

Wang *et al.* [26] proposed a privacy-preserving learning framework by formulating the problems with two mutual-information objectives, each of which is defined on a primary graph learning task and a privacy inference task. Essentially, Wang's work cast the trade-off between the primary graph learning task and a privacy inference task in an adversarial way, using an iterative minimax optimization for optimizing three neural networks. Though both of us are based on information theory, our objective cast the trade-off as a single information-theoretic objective that can be directly optimized. Our framework consists of two neural networks. Prior work on invariant representation learning has shown that adversarial training is unnecessary and might be counter-productive [27]. Moreover, Wang's work can only protect one attribute if we treat the node label as a private attribute. However, our work can deal with multiple attributes simultaneously.

The most related to our "private" graph representations is the work from Liao *et al.* [10], termed GAL (ICML21). They leveraged an adversarial learning approach to address attribute inference attacks on GNNs, by introducing a minimax game between the desired graph feature encoder and the worst-case attacker. Another related work is on compositional fairness constraints (CFC) on graph embedding proposed by Bose *et.al,* [9] (ICML19). Though CFC deals with fairness representations, the proposed approach can be generalized to privacy-preserving representation learning by simply treating the sensitive attributes as private attributes. GAL and CFC are based on adversarial training, and both are deterministic networks. Their learning performance heavily relies on the adversarial network, and they may suffer from training stability issues in practice [27]. Unlike GAL and CFC, our proposed framework *vGPF* is a variational model consisting of a GNN encoder (feature extraction network) and a prediction network. It was trained without relying on an additional adversarial network and optimized based on the proposed graph privacy funnel principle – a single information-theoretic objective that can be directly optimized over.

The other line of work on privacy-preserving GNNs is based on federated and split learning frameworks [28], [29]. Mei *et al.* leveraged structural similarity and federated learning to obfuscate nodes' sensitive information [28]. He *et at.* focused on the problem of privacy-preserving node classification by splitting the computation graph of a GNN among multiple data holders.

Privacy Funnel Problem. Privacy funnel is proposed to encode a dataset $\mathcal{D}$ into privacy-preserving mapping $\mathbf{Z}$, forming the Markov chain $\mathbf{s} \leftrightarrow \mathcal{D} \leftrightarrow \mathbf{Z}$, then share $\mathbf{Z}$ which would not reveal the private attributes $\mathbf{s}$. Formally, PF is formulated as:,

$$\begin{aligned} \arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D},\mathbf{s}}} \quad & \{I(\mathbf{s};\mathbf{Z})\} \\ \text{s.t.} \quad & \{I(\mathcal{D};\mathbf{Z}|\mathbf{s})\} \geq \upsilon, \end{aligned} \tag{1}$$

where the privacy leakage is quantified as the mutual information between the sensitive attribute $\mathbf{s}$ and privacy-preserving mapping $\mathbf{Z}$, denoted as $\{I(\mathbf{s};\mathbf{Z})\}$, and the utility requirement is modeled by a constraint υ on the average distortion measured by $\{I(\mathcal{D};\mathbf{Z}|\mathbf{s})\}$. A recent work attempts to learn private presentations based on the Lagrangians of formulated privacy optimization problem [13]. Though these existing works are all based on privacy funnel [12], [13], [30], they focus on the dataset $\mathcal{D}$ that is independent and identically distributed (IID), such as tabular data. We instead focus on privacy-preserving graph representation learning based on GNNs, where the graph-structured data is inherently non-IID.

4 GRAPH PRIVACY FUNNEL

Inheriting from the privacy funnel (PF) principle [12], the GPF objectives aim to encourage the learned graph representations to minimize the information to the sensitive attributes (privacy) and maximize the relevant information from the input graph (utility). Existing work on optimizing PF problem deals with tabular data or image data [12], [13], assuming that the training samples of the dataset are i.i.d. The non-i.i.d characteristic of graph-structured data makes optimizing the GPF problem challenging. To overcome this obstacle, we incorporate the local-dependence assumption for graph-structured data, i.e., the representation (or predictive label) of a node v_i is determined by its local computation graph denoted as $G_i^c = (V_i^c, A_i^c, X_i^c)$, where V_i^c is the node set, $A_i^c \in \{0, 1\}$ indicates the adjacency matrix, and X_i^c is the feature matrix of the computation graph. Let us take learning with a GNN encoder as an example. Essentially, a GNN learns a conditional distribution denoted as $\mathbb{P}(\mathbf{z}_i|G_i^c)$[2], where $\mathbf{z}_i$ is a random variable representing node v_i's representation. Oftentimes, the computation graph of node v_i is a L-hop subgraph, where L corresponds to the number of GNN layers. We can constrain the search space of optimal representations within the computation graph based on the local-dependence assumption, i.e., $\mathbf{z}_i \perp \mathbf{z}_j$ if $v_i \neq v_j$, leading to a more tractable GPF principle.

In what follows, we will show that though we constrain the search space of representation learning, the exact computation of GPF objectives is still challenging due to the intractability of mutual information and conditional mutual information terms [31]. We will first theoretically analyze and derive the variational bounds that can approximate the GPF objectives for optimization. Then, we introduce an instantiation for solving the GPF problem for generating graph representations that defend against attribute inference attacks in the inference stage of model partitioning settings [6], [15].

4.1 Privacy-Preserving Learning with Graph Privacy Funnel

Our objective is to obtain a GNN encoder for generating graph representations that can defend against attribute inference attacks. As commonly did in the literature [6], [9], [10], [15] , we focus on defending attribute inference attacks in the inference stage of model partitioning settings as described in Sec. 2. Specifically, the GNN encoder is trained at a trusted server. Once trained, it can be released to the data owner to generate privacy-preserving representations in the inference stage. Meanwhile, the prediction network trained with the learned representations from the GNN encoder can be released to the client for further analysis. The client should not be able to infer the private attributes according to the received representations from the data owner. However, she can use the node representations for prediction/regression (e.g., relation prediction tasks).

For training the GNN encoder, we first formulate a general probabilistic model for factoring out the protected attributes from the input graph. Specifically, the graphical model includes an observed variable $\mathbf{s}$ denotes the protected attributes that the data owner wants to remove and a continuous latent variable $\mathbf{Z}$ that models the remaining information for the client's analytic task Y. These variables naturally form two Markov Chains; they are $\mathbf{s} \leftrightarrow \mathcal{D} \leftrightarrow \mathbf{Z}$ and $Y \leftrightarrow \mathcal{D} \leftrightarrow \mathbf{Z}$. Inspired by the PF principle proposed by Makhdoumi *et at.*, we can formally define our training objective as follows.

$$\begin{aligned} \arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D},\mathbf{s}}} \quad & \{I(\mathbf{s};\mathbf{Z})\} + I(\mathcal{D};\mathbf{Z}|\mathbf{s},Y) \\ \text{s.t.} \quad & \{I(Y;\mathbf{Z}|\mathbf{s})\} \geq \upsilon. \end{aligned} \tag{2}$$

The first term $I(\mathbf{s};\mathbf{Z})$ characterizes the information the learned representations $\mathbf{Z}$ retained about the sensitive attributes $\mathbf{s}$. The second term $I(\mathcal{D};\mathbf{Z}|\mathbf{s},Y)$ can be interpreted as, given the prediction task, the learned representations $\mathbf{Z}$ should capture the minimum sufficient relevant information within the input graph related to the target label Y. The constraint term ensures the prediction performance of the target task, i.e., the data utility of generated representations for predicting Y. We can solve formula (2) with Lagrangian relaxation. In general, the Lagrangian is a widely used methodology for characterizing the trade-off between the optimized function and the constraints on the search space [32].

$$\arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D},\mathbf{s}}} I(\mathbf{s};\mathbf{Z}) + I(\mathcal{D};\mathbf{Z}|\mathbf{s},Y) - \upsilon I(Y;\mathbf{Z}|\mathbf{s}). \tag{3}$$

Proposition 4.1. Solving formula (3) is equivalent to minimize $\mathcal{L}$, where $\beta = \upsilon + 1$ and

$$\mathcal{L} = I(\mathcal{D};\mathbf{Z}) - \beta I(Y;\mathbf{Z}|\mathbf{s}). \tag{4}$$

Proof:

$$\arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} I(\mathbf{s};\mathbf{Z}) + I(\mathcal{D};\mathbf{Z}|\mathbf{s},Y) - \upsilon I(Y;\mathbf{Z}|\mathbf{s}) \tag{5}$$

$$= \arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} H(\mathbf{s}) + H(\mathbf{Z}) - H(\mathbf{s},\mathbf{Z}) + \tag{6}$$

$$H(\mathcal{D}|\mathbf{s},Y) + H(\mathbf{Z}|\mathbf{s},Y) - H(\mathcal{D},\mathbf{Z}|\mathbf{s},Y) - \upsilon I(Y;\mathbf{Z}|\mathbf{s}) \tag{7}$$

$$= \arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} H(\mathbf{Z}) + H(\mathcal{D},\mathbf{s},Y) - H(\mathcal{D},\mathbf{Z},\mathbf{s},Y) - \upsilon I(Y;\mathbf{Z}|\mathbf{s}) \tag{8}$$

$$= \arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} H(\mathbf{Z}) + H(\mathcal{D}) - H(\mathcal{D},\mathbf{Z}) - \upsilon I(Y;\mathbf{Z}|\mathbf{s}) \tag{9}$$

$$= \arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} H(\mathbf{Z}) + H(\mathcal{D}) - H(\mathcal{D},\mathbf{Z}) - H(Y,\mathbf{s}) - \tag{10}$$

$$H(\mathbf{Z},\mathbf{s}) + H(\mathbf{s}) + H(Y,\mathbf{Z},\mathbf{s}) - \upsilon I(Y;\mathbf{Z}|\mathbf{s}) \tag{11}$$

$$= \arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} I(\mathcal{D};\mathbf{Z}) - I(Y;\mathbf{Z}|\mathbf{s}) - \upsilon I(Y;\mathbf{Z}|\mathbf{s}) \tag{12}$$

$$= \arg \inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} I(\mathcal{D};\mathbf{Z}) - (\upsilon + 1) I(Y;\mathbf{Z}|\mathbf{s}). \tag{13}$$

Without loss of generality, $H(\cdot)$ denotes the information entropy of variables. In particular, formula (8) to formula (11) are derived based on the Markov chain's property. The Markov chains are $\mathbf{s} \leftrightarrow \mathcal{D} \leftrightarrow \mathbf{Z}$ and $Y \leftrightarrow \mathcal{D} \leftrightarrow \mathbf{Z}$.

With **Proposition 4.1**, we can minimize Eq. (4) to transform the input graph into privacy-preserving mapping with respect to prediction Y. Nevertheless, exact computation of $I(\mathcal{D};\mathbf{Z})$ and $I(Y;\mathbf{Z}|\mathbf{s})$ is still intractable. Therefore, we derive two variational bounds on these two terms as shown in **Proposition 4.2** and **4.3** ; the variation methods are widely used in model optimization [13], [33].

Proposition 4.2. (The lower bound of $I(Y;\mathbf{Z}|\mathbf{s})$). For any probabilistic distribution functions $\mathbb{P}(Y|\mathbf{Z},\mathbf{s})$ and $\mathbb{Q}(Y|\mathbf{s})$, we have

[2]For ease of notation, we use $\mathbb{P}(\mathbf{z}_i|\mathcal{D})$ in the entire paper though $\mathbf{z}_i$ is determined by its surrounding computation graph G_i^c.

$$I(Y;\mathbf{Z}|\mathbf{s}) \geq 1 + \mathbb{E}_{\mathbb{P}_{Y,\mathbf{Z},\mathbf{s}}}\left[\log\left(\frac{\mathbb{P}(Y|\mathbf{Z},\mathbf{s})}{\mathbb{Q}(Y|\mathbf{s})}\right)\right] \tag{14}$$

$$+ \mathbb{E}_{\mathbb{P}(Y|\mathbf{s})\mathbb{P}(\mathbf{Z})}\left[\frac{\mathbb{P}(Y|\mathbf{Z},\mathbf{s})}{\mathbb{Q}(Y|\mathbf{s})}\right] \tag{15}$$

$$\geq \mathbb{E}_{\mathbb{P}_{Y,\mathbf{Z},\mathbf{s}}}\left[\log\left(\frac{\mathbb{P}(Y|\mathbf{Z},\mathbf{s})}{\mathbb{Q}(Y|\mathbf{s})}\right)\right] \tag{16}$$

$$= \mathbb{E}_{\mathbb{P}_{Y,\mathbf{Z},\mathbf{s}}}[\log \mathbb{P}_{Y|\mathbf{Z},\mathbf{s}}] - \mathbb{E}_{\mathbb{P}_{Y,\mathbf{Z},\mathbf{s}}}[\log \mathbb{Q}_{Y|\mathbf{s}}]. \tag{17}$$

We follow the work [13] and use the Nguyen, Wainright & Jordan's bound $I_{\mathbf{NWJ}}$ [34], [35]:

Lemma 4.1. For any two random variables X_1, X_2 and any function $g : g(X_1, X_2) \in \mathbb{R}$, we can have

$$I(X_1, X_2) \geq \mathbb{E}\left[g(X_1, X_2)\right] - \tag{18}$$

$$\mathbb{E}_{\mathbb{P}(X_1)\mathbb{P}(X_2)}\left[\exp\left(g(X_1, X_2) - 1\right)\right]. \tag{19}$$

We first use **Lemma 4.1** to derive the formula (14) and (15) of $I(Y;\mathbf{Z}|\mathbf{s})$ by plugging in $g(Y|\mathbf{Z},\mathbf{s}) = 1 + \log\frac{\mathbb{P}(Y|\mathbf{Z},\mathbf{s})}{\mathbb{Q}(Y|\mathbf{s})}$.

With **Proposition 4.2**, $I(Y;\mathbf{Z}|\mathbf{s})$ can be further bounded by formula (20) due to the non-negativity of expectation term $\mathbb{E}_{\mathbb{P}(Y|\mathbf{s})\mathbb{P}(\mathbf{Z})}\left[\frac{\mathbb{P}(Y|\mathbf{Z},\mathbf{s})}{\mathbb{Q}(Y|\mathbf{s})}\right]$.

$$I(Y;\mathbf{Z}|\mathbf{s}) \geq \mathbb{E}_{\mathbb{P}_{Y,\mathbf{Z},\mathbf{s}}}\left[\log\left(\frac{\mathbb{P}(Y|\mathbf{Z},\mathbf{s})}{\mathbb{Q}(Y|\mathbf{s})}\right)\right]. \tag{20}$$

Proposition 4.3. (The upper bound of $I(\mathcal{D};\mathbf{Z})$). For any probabilistic distribution functions $\mathbb{P}(\mathbf{Z}|\mathcal{D})$ and $\mathbb{Q}(\mathbf{Z})$, we have

$$I(\mathcal{D};\mathbf{Z}) \leq \mathbb{D}_{\mathbb{KL}}\left(\mathbb{P}_{\mathbf{Z}|\mathcal{D}}||\mathbb{Q}_{\mathbf{Z}}\right). \tag{21}$$

Put together, we can have the following upper bound to minimize:

$$\mathcal{L} \approx \mathbb{D}_{\mathbb{KL}}\left(\mathbb{P}_{\mathbf{Z}|\mathcal{D}}||\mathbb{Q}_{\mathbf{Z}}\right) - \beta\mathbb{E}_{\mathbb{P}_{Y,\mathbf{Z},\mathbf{s}}}\left[\log\left(\frac{\mathbb{P}(Y|\mathbf{Z},\mathbf{s})}{\mathbb{Q}(Y|\mathbf{s})}\right)\right]. \tag{22}$$

Generalize to Unsupervised Privacy-Preserving Learning. There is a situation where the downstream tasks are unknown. The server needs to encode the input graph $\mathcal{D} = (\mathbf{A}, \mathbf{X})$ into the privacy-preserving node representations $\mathbf{Z}$ in the unsupervised mode. Specifically, we can define the privacy-preserving graph mapping in the unsupervised mode where the privacy leakage is quantified as the mutual information between the sensitive attribute $\mathbf{s}$ and privacy-preserving mapping $\mathbf{Z}$, denoted as $I(\mathbf{s},\mathbf{Z})$, and the data utility can be measured by the mutual information between the privacy-preserving mapping $\mathbf{Z}$ and the reconstructed data, e.g., $\mathbf{A}$ for graph-structured data [11]. This mapping process forms a Markov chain $\mathbf{s} \leftrightarrow \mathcal{D} \leftrightarrow \mathbf{Z}$. Accordingly, we can formally define the objective as follows.

$$\arg\inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D},\mathbf{s}}} I(\mathbf{s};\mathbf{Z}) - \upsilon I(\mathcal{D};\mathbf{Z}|\mathbf{s}). \tag{23}$$

Similarly, we can derive the final minimization objective for learning privacy-preserving graph mapping in the unsupervised manner, formulated as:

$$\mathcal{L}_{un} \approx \mathbb{D}_{\mathbb{KL}}\left(\mathbb{P}_{\mathbf{Z}|\mathcal{D}}||\mathbb{Q}_{\mathbf{Z}}\right) - \beta\mathbb{E}_{\mathbf{P}_{\mathcal{D},\mathbf{Z},\mathbf{s}}}\left[\log\left(\frac{\mathbb{P}_{\mathcal{D}|\mathbf{Z},\mathbf{s}}}{\mathbb{Q}_{\mathcal{D}|\mathbf{s}}}\right)\right]. \tag{24}$$

Proposition 4.4. Solving formula (24) is equivalent to minimize $\mathcal{L}_{un}$, where $\beta = \upsilon + 1$ and

$$\mathcal{L}_{un} = I(\mathcal{D};\mathbf{Z}) - \beta I(\mathcal{D};\mathbf{Z}|\mathbf{s}). \tag{25}$$

Proof.

$$\arg\inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} I(\mathbf{s};\mathbf{Z}) - \upsilon I(\mathcal{D};\mathbf{Z}|\mathbf{s}) \tag{26}$$

$$= \arg\inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} I(\mathcal{D};\mathbf{Z}) - I(\mathcal{D};\mathbf{Z}|\mathbf{s}) - \upsilon I(\mathcal{D};\mathbf{Z}|\mathbf{s}) \tag{27}$$

$$= \arg\inf_{\mathbf{P}_{\mathbf{Z}|\mathcal{D}}} I(\mathcal{D};\mathbf{Z}) - (\upsilon + 1) I(\mathcal{D};\mathbf{Z}|\mathbf{s}). \tag{28}$$

Similarly, we can derive the lower bound of $I(\mathcal{D};\mathbf{Z}|\mathbf{s}))$ as follows:

Proposition 4.5. (The lower bound of $I(\mathcal{D};\mathbf{Z}|\mathbf{s})$).

$$I(\mathcal{D};\mathbf{Z}|\mathbf{s}) \geq \mathbb{E}_{\mathbf{P}_{\mathcal{D},\mathbf{Z},\mathbf{s}}}\left[\log\left(\frac{\mathbb{P}_{\mathcal{D}|\mathbf{Z},\mathbf{s}}}{\mathbb{Q}_{\mathcal{D}|\mathbf{s}}}\right)\right]. \tag{29}$$

Proposition 4.5 leads us the final minimization objective for learning privacy-preserving graph mapping in the unsupervised manner, formulated as:

$$\mathcal{L}_{un} \approx \mathbb{D}_{\mathbb{KL}}\left(\mathbb{P}_{\mathbf{Z}|\mathcal{D}}||\mathbb{Q}_{\mathbf{Z}}\right) - \beta\mathbb{E}_{\mathbf{P}_{\mathcal{D},\mathbf{Z},\mathbf{s}}}\left[\log\left(\frac{\mathbb{P}_{\mathcal{D}|\mathbf{Z},\mathbf{s}}}{\mathbb{Q}_{\mathcal{D}|\mathbf{s}}}\right)\right]. \tag{30}$$

4.2 The Instantiation for Solving Graph Privacy Funnel

We now discuss how to compute the final objective with formula (22) in practice. To estimate $I(\mathcal{D};\mathbf{Z})$, we need to model $\mathbb{P}(\mathbf{Z}|\mathcal{D})$ and $\mathbb{Q}(\mathbf{Z})$. For this purpose, we use a GNN encoder parameterized by θ and we take a mixture of Gaussians with learnable parameters [36] respectively. Specifically, the Gaussian prior for each node v_i is formulated as $\mathbf{z}_i \sim \sum_{j=1}^{f'} \mathbf{w}_j \mathbf{Guassian}(\mu_{0,j}, \sigma_{0,j}^2)$, where w_j, $\mu_{0,j}$ and $\sigma_{0,j}$ are learnable parameters shared by all nodes and f' is the dimension of $\mathbf{z}_i$. With the local-dependence assumption, we can have $\mathbb{Q}(\mathbf{Z}) = \Pi_{v_i \in V}\mathbb{Q}(\mathbf{z}_i)$. The stochastic encoder takes the form of $\mathbb{P}_\theta(\mathbf{z}_i|\mathcal{D}) = \mathcal{N}(\mathbf{z}_i|\Psi^{\mu}(\mathcal{D}), \Psi^{\sigma^2}(\mathcal{D}))$ by using the reparameterization trick, where Ψ is an GNN encoder and its output is a latent vector in $\mathbb{R}^{2f'}$. The first f' outputs encode μ and the remaining f' encode σ after a softplus transform.

To estimate $I(Y;\mathbf{Z}|\mathbf{s})$, we adopt a decoder parameterized by ϕ that admits two random variables: the node representations $\mathbf{z}_i$ and the sensitive attributes $\mathbf{s}$. This decoder enables feasible backpropagation on the GNN encoder to discard the sensitive information and retain the maximal information for predicting Y without training any additional adversarial networks. $\mathbb{Q}_{Y|\mathbf{s}}$ in formula (14) can be inferred from the data. In other words, $\mathbb{Q}_{Y|\mathbf{s}}$ in formula (14) does not depend on the parameterization process; we can discard this term from the optimization. Therefore, we only need to model $\mathbb{P}_\phi(Y|\mathbf{Z},\mathbf{s})$ for evaluating $I(Y;\mathbf{Z}|\mathbf{s})$. For this purpose, we can simply set $\mathbb{P}_\phi(Y|\mathbf{Z},\mathbf{s}) = \Pi_{i \in V}\mathbb{P}_\phi(y_i|(\mathbf{z}_i||\mathbf{s}))$, where $||$ is the concatenation operator. Then, $I(Y;\mathbf{Z}|\mathbf{s})$ reduces to the cross-entropy loss of the target task by ignoring the constants. All together, we can describe our proposed variational model as Fig. 1(b). The model parameters will be jointly optimized with our GPF objective. Once trained, the data owner can generate privacy-preserving node representations through the trained GNN encoder and release them to the client for downstream analysis. In Sec. 5, we will show that our

proposed variational model is compatible with various GNN encoder architectures.

Discussions. We show that the proposed approach can readily generalize to unsupervised privacy-preserving graph representation learning. Nevertheless, when the sensitive attributes $\mathbf{s}$ are correlated with the downstream tasks (e.g., relation prediction tasks), removing them from the representations ((to be disclosed) without considering the target task can be harmful to the task performance. In other words, it might be hard to guarantee the service utility if learning privacy-preserving representations without specifying the ground-truth values Y of the target task. Therefore, it is preferable to learn privacy-preserving graph representations in a supervised way. Thus, in this paper, **we focus on evaluating the effectiveness of *vGPF* in supervised graph representation learning.**

One may find out that our GPF objective and graph information bottleneck (GIB) [14] are both information-theoretic principles for learning graph representations. However, GIB and GPF are different regarding the learning objectives. GIB learns graph representations explicitly robust against adversarial attacks, whereas GPF aims to learn graph representations robust against attribute inference attacks. The model architectures are also different; our decoder is designed to admit the representations and sensitive attributes for guiding the GNN encoder to discard the sensitive information and retain the maximal information for predicting Y.

5 Experimental Studies

5.1 Datasets and Experimental Setup

Datasets. We performed experiments using 6 datasets that were previously used by Liao *et.al,* [10] and Bose *et.al,* [9]. Our experiments were conducted over various GNN encoder architectures, e.g., GCN [18], ChebNet [17], CompGCN [37]. These relation prediction benchmarks are from different application domains, including recommender systems on MovieLens-1M [38], citation networks on Cora, CiteSeer and Pubmed, and knowledge graphs on FB15K-237 and WN18RR [39].

TABLE 1
The statistics of used datasets, corresponding evaluation metrics for the target tasks, and the metrics of adversarial attribute inference attacks. f is the dimension of node-level non-sensitive attributes (i.e., the input node feature dimension).

Dataset	#of nodes $\lvert V\rvert$	#of edges $\lvert E\rvert$	f	Task Metrics	Adversary Metrics
Cora	2,708	5,429	1,433	AUC	Macro-F1
CiteSeer	3,327	4,552	3,703	AUC	Macro-F1
Pubmed	19,717	44,324	500	AUC	Macro-F1
ML-1M	9,940	1,000,209	1 (id)	RMSE	Macro-F1/AUC
FB15K-237	14,940	168,618	1 (id)	MRR	Macro-F1
WN18RR	40,943	173,670	1 (id)	MRR	Macro-F1

Baselines. We compare our approach against the the-state-of-the-art solutions for obfuscating information from GNNs, including two realizations, i.e., GAL-W and GAL-TV [10]. In addition, CFC [9] can essentially deal with privacy-preserving graph representation learning if we treat sensitive attributes $\mathbf{s}$ as the private factors. We also take the state-of-art method for fair graph embedding (CFC) as our baseline [3]. Unless otherwise stated, for GAL, we use GAL-W as the representative baseline[4], and we set all the hyperparameters of the baselines as reported in the corresponding papers.

We evaluate the effectiveness of *vGPF* from the following aspects. *First*, we test the effectiveness of *vGPF* across various tasks and GNN encoders, such as GCN, ChebNet, CompGCN, etc. This component is to check if *vGPF* is a general framework that can be compatible with various tasks with different GNN encoder architectures. *Second*, as the proposed approach can control the trade-off between privacy and accurate representations, we explore the trade-off that can be achieved with *vGPF* by tuning the value of β. *Third*, we examine the information leakage of generated embedding to confirm that *vGPF* indeed can defend against attribute inference attacks by various types of adversaries. In particular, we further check *vGPF* can generate privacy-preserving representations that can enforce privacy on multiple sensitive attributes simultaneously. In this component, we compare *vGPF* with CFC proposed by Bose *et.al,* [9].

Evaluation Metrics. The evaluation for *vGPF* is geared towards two fronts; removing information about the sensitive attributes $\mathbf{s}$ and maintaining the predictive accuracy of Y. The evaluating metric of predictive performance varies in different learning tasks. For example, the metric for the recommender system on MovieLens-1M is RMSE, while the one for knowledge graphs is MRR. The statistics of used datasets and the performance evaluation metric for each target task are in Table 1. Moreover, we assume the adversaries in all experiments have access to the node representations and corresponding attribute labels for training the attack model. They perform node classification for inferring sensitive attributes, where the performance is measured on the held-out test set. We use the accuracy of attribute inference as the quantification of information leakage (see the adversary metrics in Table 1). Unless otherwise stated, the inference classifiers are instantiated with multi-layer perceptrons (MLPs) with LeakyReLU (See Appendix for further description) [5]. We report average performance (mean and standard deviation) over five runs. A further detailed description of corresponding graph learning tasks and implementation details [5] (e.g., model architectures, the training details of GNN encoders and attack models.) are elaborated in **Implementation Details** of Appendix.

5.2 Results and Discussion

Performance Comparisons on MovieLens-1M. We start with experiments on the MovieLens-1M recommender system benchmark. Following GAL and CFC, we treat the user features, including age, gender, and occupation, as sensitive attributes. *Original* refers to the model performance of the original task without enforcing resistance against attribute inference attack. Though these features are not input features for predicting the rating between the users and movies, they are highly correlated with the rating value Y; thus,

[3]We use the source code released by the authors.

[4]As GAL-TV and GAL-W show similar performance, we put the results for GAL-TV in Appendix.

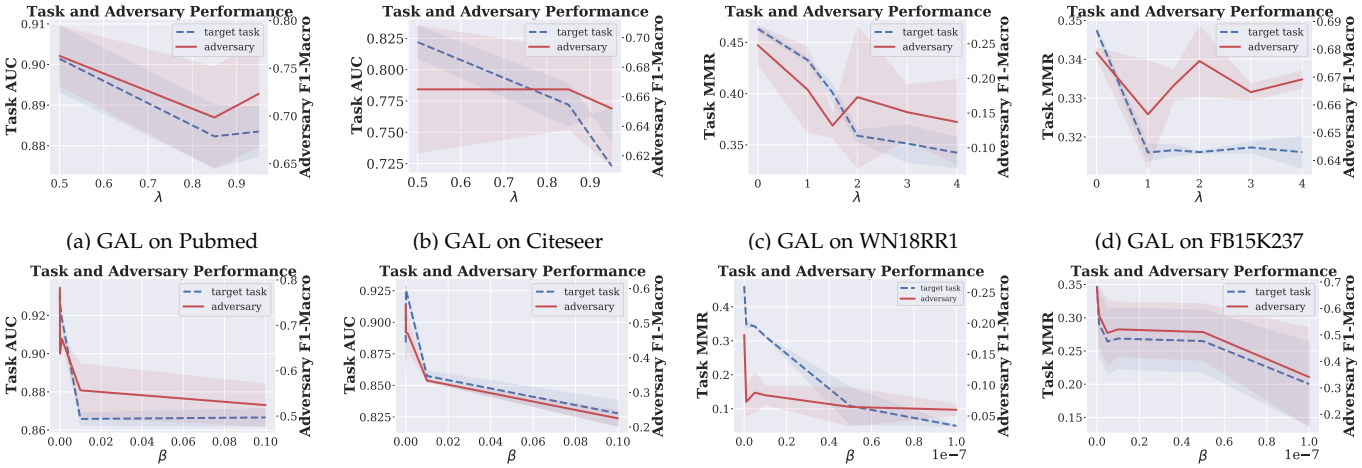
[5]The source code can be found in Supplementary Material.

TABLE 2
Performance comparisons on MovieLens-1M. The data utility on MovieLens-1M is measured by the rating prediction accuracy with RMSE, and the sensitive information leakage is quantified by the attribute inference accuracy of the adversary with AUC/Macro-F1.

Attribute	Rating Prediction RMSE (smaller is better)					Attribute Inference Accuracy (smaller is better)				
	Original	CFC	GAL-TV	GAL-W	*vGPF*	Original	CFC	GAL-TV	GAL-W	*vGPF*
Gender	0.870±0.003	0.927±0.004	0.928±0.016	0.981±0.082	**0.905±0.003**	0.753±0.011	0.524±0.018	0.585±0.103	0.501±0.036	**0.499±0.008**
Age	0.870±0.003	0.988±0.001	0.876±0.004	1.213±0.022	**0.871±0.000**	0.192±0.010	0.112±0.007	0.110±0.022	0.095±0.022	**0.086±0.002**
Occupation	0.870±0.003	0.904±0.003	0.983±0.059	0.911±0.005	**0.904±0.001**	0.048±0.005	0.039±0.009	0.018±0.004	0.018±0.006	**0.015±0.001**

TABLE 3
Performance on other benchmark datasets. β is the tunable trade-off parameter. The statistics are represented as {attribute inference accuracy}/{the target task accuracy} (Corresponding performance metrics are presented in Table 1).

CORA	β	0	0.0001	0.001	0.01	0.1
	ChebNet	0.759±0.011/0.918±0.006	0.62±0.017/0.937±0.007	0.648±0.013/0.933±0.006	0.388±0.016/0.88±0.012	0.315±0.011/0.873±0.008
	GCN	0.763±0.016/0.921±0.006	0.700±0.017/0.930±0.006	0.687±0.012/0.920±0.007	0.405±0.016/0.884±0.012	0.348±0.038/0.885±0.010
PUBMED	ChebNet	0.783±0.005/0.931±0.004	0.638±0.015/0.926±0.002	0.670±0.011/0.918±0.004	0.557±0.058/0.866±0.003	0.524±0.047/0.867±0.005
	GCN	0.773±0.007/0.964±0.001	0.639±0.029/0.948±0.001	0.656±0.013/0.942±0.001	0.467±0.046/0.890±0.000	0.399±0.012/0.905±0.003
CITESEER	ChebNet	0.554±0.054/0.884±0.022	0.473±0.078/0.925±0.004	0.470±0.050/0.920±0.003	0.334±0.004/0.857±0.004	0.225±0.023/0.828±0.010
	GCN	0.616±0.009/0.902±0.003	0.556±0.026/0.917±0.006	0.533±0.030/0.896±0.010	0.334±0.024/0.861±0.010	0.279±0.012/0.853±0.006
	β	0	$1*10^{-9}$	$5*10^{-9}$	$1*10^{-8}$	$5*10^{-8}$
$WN18RR_1$	CompGCN	0.181±0.080/0.462±0.001	0.073±0.023/0.349±0.009	0.088±0.032/0.343±0.003	0.084±0.015/0.313±0.002	0.065±0.031/0.109±0.054
$WN18RR_2$	CompGCN	0.754±0.084/0.462±0.001	0.696±0.007/0.252±0.168	0.703±0.003/0.344±0.004	0.702±0.003/0.325±0.002	0.693±0.014/0.133±0.005
FB15K-237	β	0	10^{-9}	10^{-8}	10^{-7}	10^{-6}
	CompGCN	0.675±0.003/0.348±0.001	0.574±0.113/0.291±0.048	0.508±0.121/0.264±0.049	0.521±0.103/0.269±0.046	0.511±0.107/0.265±0.047



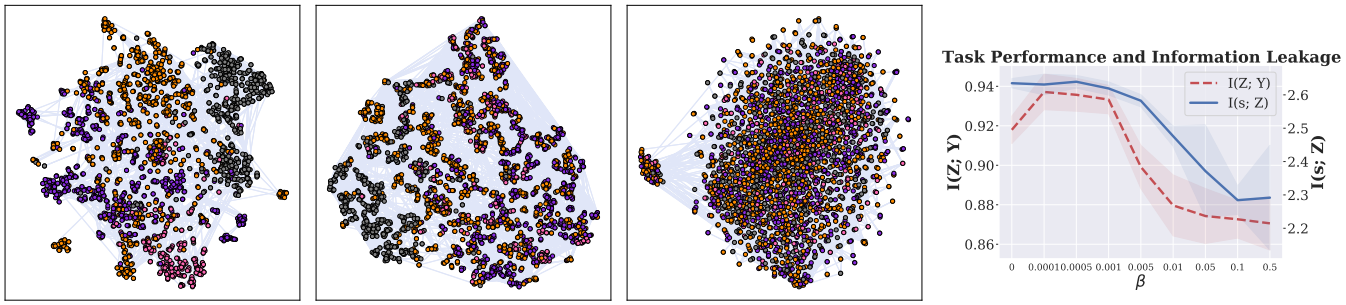
(a) GAL on Pubmed (b) GAL on Citeseer (c) GAL on WN18RR1 (d) GAL on FB15K237

(e) *vGPF* on Pubmed (f) *vGPF* on Citeseer (g) *vGPF* on WN18RR1 (h) *vGPF* on FB15K237

Fig. 2. Comparisons with the representative and state-of-the-art baseline, GAL (ICML21). Generally, *vGPF* exhibits more stable performance in the target task (with smaller variance) compared with GAL.

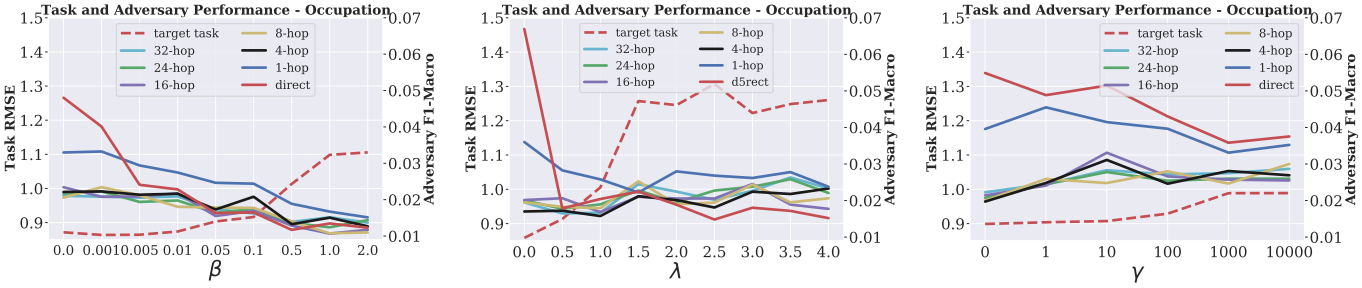
removing them from the representations is challenging. Table 2 shows the performance evaluations on MovieLens-1M. Our approach surpasses the existing baselines: it exhibits better resistance against attribute inference attacks while sacrificing less target task performance. *vGPF* also shows more stable predictive performance in the target task – with smaller variance as compared to the adversarial training-based approaches. These results are consistent with our claims. We also observe that the effects of different sensitive attributes vary; removing gender is harder than occupation, indicating gender is more correlated with the ratings.

Trade-Off Tuning with β over Various GNN Encoders. Table 3 illustrates the effectiveness of *vGPF* across various GNN encoder backbones, including GCN, ChebNet, and CompGCN. We show that *vGPF* allows tuning with β to quantify the trade-off between data utility and the privacy leakage by testing the performance over a wide range of



(a) $\beta = 0$ (b) $\beta = 0.005$ (c) $\beta = 0.05$ (d) Information measurement.

Fig. 3. 3a, 3b and 3c depict **t-SNE visualizations of Cora** under $\beta = 0$, $\beta = 0.005$, and $\beta = 0.05$, respectively. Node colors represents node classes. 3d depicts the utility-privacy trade-off by information measurement. We can observe that the sensitive information included in the representations is indeed reduced along with the increase of β.



(a) *vGPF* (b) GAL-W (c) CFC

Fig. 4. **Robustness against occupation inference to various attribute inference attacks.** The direct attack and 1-hop attack on all methods perform the best (best viewed in color).

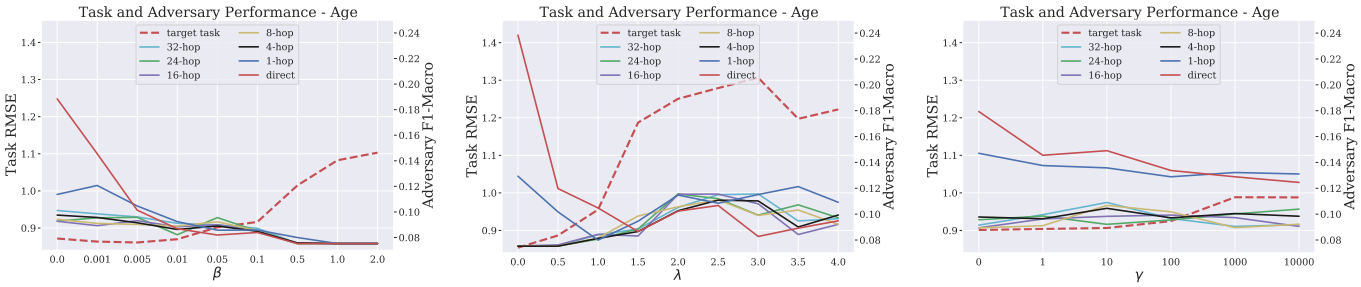
β on all benchmarking datasets. Specifically, a larger value of β can defend against attribute inference attacks better (with lower values of AUC/Macro-F1). However, increasing β degenerates the predictive accuracy of the target task.

Comparisons with GAL. GAL [10] (ICML21) is the state-of-the-art baseline based on adversarial training for information obfuscation of GNNs. Both *vGPF* and GAL all contain a tunable parameter that can adjust the trade-off between the target task performance and the information leakage, i.e., β in *vGPF* and λ in GAL. Fig. 2 illustrates the trade-off tuning comparisons over various benchmark datasets. We use the same GNN encoder for fair comparisons in this setting, i.e., ChebNet. Both *vGPF* and GAL enjoy a similar trend in terms of tuning the trade-off parameters. Specifically, increasing the tunable parameters, i.e., β/λ, damages the model's accuracy on the target task while the accuracy of attribute inference decreases. We observe that *vGPF* exhibits more stable and better predictive performance (with smaller variance) compared with GAL in the target task. This may indicate that adversarial training-based approaches have stability issues and can reach sub-optimum predictive performance.

Visualization. Fig. 3 depicts the visualized t-SNE of *vGPF* for removing sensitive attributes (node labels for Cora) from the learned representations. Specifically, Fig. 3a, 3b, and 3c visualize the node representations of Cora with $\beta = 0$, $\beta = 0.005$, and $\beta = 0.05$, respectively. We observe that the sensitive attributes (node labels) are better mixed when β is higher — nodes belonging to different classes (in different colors) are more indistinguishable. In Fig. 3d, we measure the target task performance with learned representations and the information leakage by evaluating corresponding mutual information [31] under various β. Specifically, the information leakage is measured by mutual information $I(\mathbf{s}; \mathbf{Z})$, and the target task performance is quantified by $I(\mathbf{Z}; Y)$. We can observe that with the increase of β, the information leakage decreases (better privacy), and the performance of the target task degenerates as well.

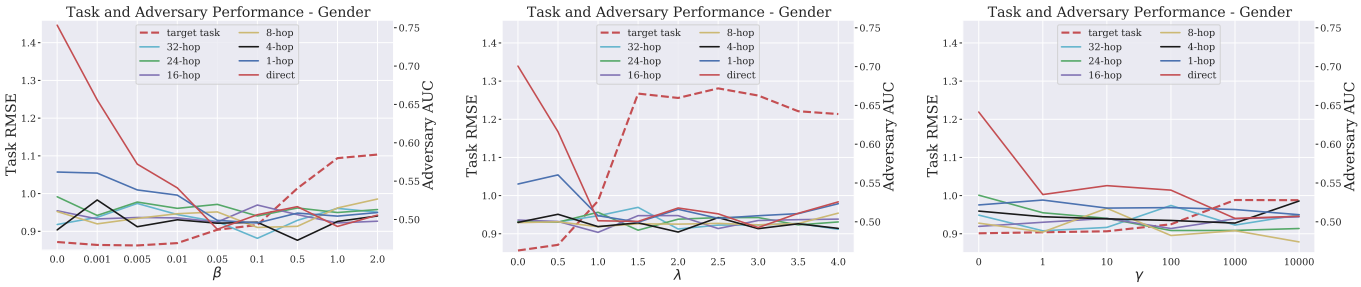
Robustness to Attribute Inference Attacks. *Direct attack* is defined as inferring attributes based on the target node's representation. In contrast, the n-hop attack is based on the representations of neighbors that are n-hop away from the target node. We first consider the worst-case adversary who has access to all the representations and performs direct attacks on the target nodes. Moreover, we use the n-hop attack to evaluate the vulnerability of GNN caused by its neighborhood-aggregation scheme, as did in [10]. Specifically, except for the direct attacks, we further assess the robustness of generated representations in the following scenarios: the adversary has access to the representations of neighbors that are n-hop away from the target nodes, where $n = 1, 4, 8, 16, 24, 32$, respectively.

Fig. 4 plots the performance comparisons of three methods on removing the occupation information from the learned representations. As expected, the direct attack and 1-hop attack on all methods perform the best among all adversarial attacks, particularly with low values of β, λ, and γ. In general, with the increase of the trade-off parameters,



(a) *vGPF* (b) GAL-W (c) CFC

Fig. 5. **Robustness against age inference to various attribute inference attacks.** The direct attack and 1-hop attack on all methods perform the best (best viewed in color).



(a) *vGPF* (b) GAL-W (c) CFC

Fig. 6. **Robustness against gender inference to various attribute inference attacks.** The direct attack and 1-hop attack on all methods perform the best (best viewed in color).

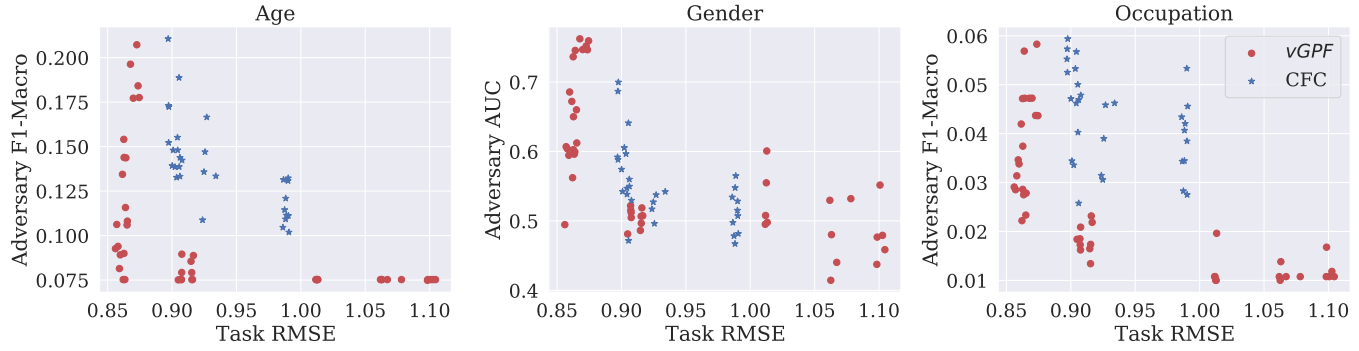
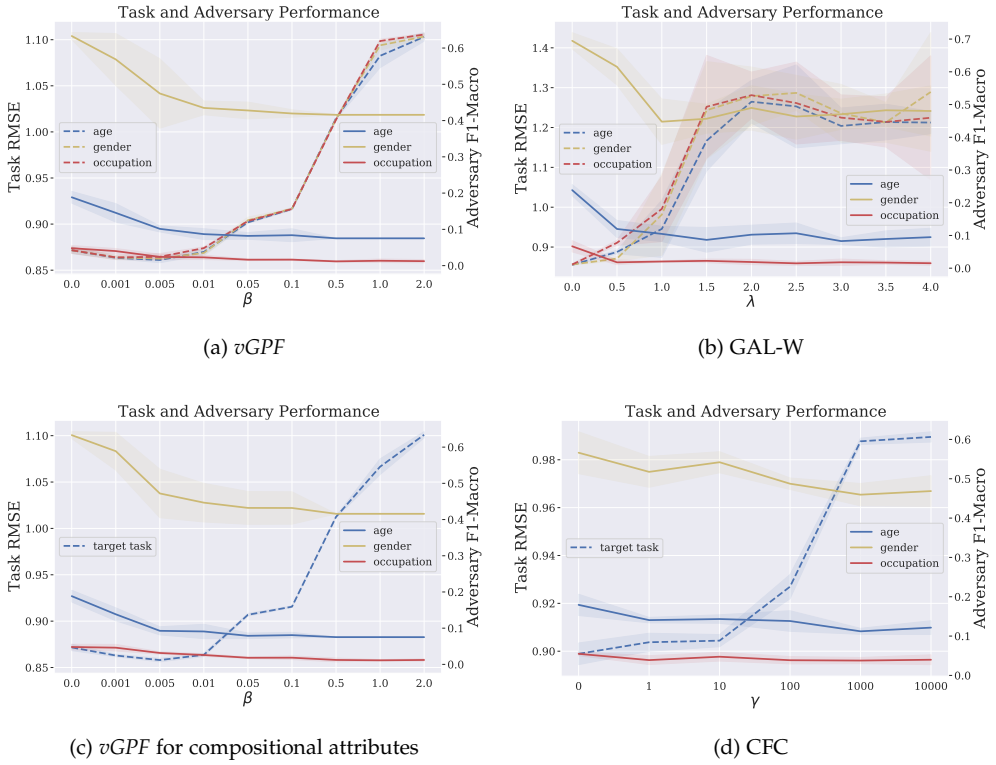


Fig. 7. Utility-Privacy trade-off plot for removing multiple attributes on MovieLens-1M. To compare with the compositional approach CFC, we train our variational model with compositional attributes $\mathbf{s}$ consisting of age, gender and occupation (by concatenation operator). We then use three adversarial classifiers instantiated with MLPs for inferring age, gender and occupation, respectively. Points located at the left bottom are better.

the performance of the target task degenerates while the information leakage in terms of adversary performance decreases. Interestingly, *vGPF* appears to have better task performance yet have lower information leakage. For example, *vGPF* can obtain 0.904 task RMSE and 0.016 Macro-F1 with direct attack ($\beta = 0.05$). In contrast, with similar task performance, the adversary of GAL-W can achieve 0.018 ($\lambda = 0.5$), and CFC gets 0.039 ($\gamma = 1$).

Fig. 5 and Fig. 6 plot the performance comparisons of three methods on removing the age and gender information from the learned representations, respectively. We evaluate corresponding information leakage through attribute inference attacks. As expected, the direct attack and 1-hop attack perform the best among all adversarial attacks, particularly with low values of β, λ, and γ. In general, with the increase of the trade-off parameters, the performance of the target task degenerates (low data utility) while the information leakage in terms of adversary performance decreases (better privacy). As shown in Fig. 5 and Fig. 6, we can observe that among all privacy-preserving representation learning methods, *vGPF* appears to have better task performance yet have lower information leakage.

Performance on removing multiple sensitive attributes. Both CFC and *vGPF* can enforce privacy (remove sensitive



(a) *vGPF* (b) GAL-W

(c) *vGPF* for compositional attributes (d) CFC

Fig. 8. **Trade-off parameter tuning.** *vGPF* offers a more stable task performance overall. And *vGPF* can enforce privacy over multiple attributes simultaneously and provide comparable performance as compared to baselines.

information) over multiple attributes. In this comparison, we train *vGPF* using compositional attributes $\mathbf{s}$ consisting of age, gender, and occupation. These attributes are combined using the concatenation operator and as the input of the decoder. We then evaluate the trade-off performance with various tunable parameters, i.e., β of *vGPF* and γ of CFC. Concretely, we use three adversarial classifiers instantiated with MLPs for inferring age, gender, and occupation, respectively. Fig. 7 plots the trade-off performance comparisons. In general, *vGPF* exhibits better performance. Points located at the left bottom are better.

Fig. 8 plots the performance of *vGPF*, GAL-W, and CFC on MovieLens-1M under different tunable parameters, with 95% confidence interval over 5 runs. In particular, Fig. 8a and 8b show that *vGPF* can generate node representations that have less information leakage for enforcing privacy on one attribute at a time, as compared to GAL-W. Fig. 8c and 8d show that *vGPF* can enforce privacy over multiple attributes in one go and it can provide comparable performance as compared to the state of the arts. More importantly, *vGPF* offers the best stability in terms of the task performance among all methods.

6 Conclusion and Future Directions

We introduce a new variational framework *vGPF* for learning privacy-preserving graph representations that use the principle of graph privacy funnel. *vGPF* has several advantages: it is agnostic to the GNN encoder architecture, it can generate privacy-preserving graph representations across various domains with better and more stable predictive performance, enjoys high flexibility in the sense that it can enforce privacy over arbitrary combinations of sensitive attributes in a plug-and-play manner. We find that our approach surpasses recently proposed adversarial approaches.

There are many possible directions for future work, such as putting the GPF objective with other graph variational models. In addition, our proposed approach is based on the privacy funnel—an information-theoretical principle and does not provide a rigorous mathematical guarantee for ensuring privacy like the notion of differential privacy. Further exploration of how our proposed approach connects to differential privacy could be a potential direction to illustrate the effectiveness of the privacy funnel principle for ensuring privacy beyond empirical evidence. We hope our study establishes the foundations of using GNN encoders in privacy-sensitive applications.

Acknowledgments

This work was supported in part by the Research Institute for Artificial Intelligence of Things, the Hong Kong Polytechnic University, in part by the Hong Kong (HK) Research Grant Council (RGC) General Research Fund under Grant PolyU 15208222, in part by NSFC Young Scientist Fund under Grant PolyU A0040473, and in part by HK RGC Theme-based Research Scheme, under Grants T43- 513/23-N and T41-603/20-R.

REFERENCES

[1] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proc. SIGKDD*. ACM, 2018.

[2] Dawei Cheng, Yi Tu, Zhen-Wei Ma, Zhibin Niu, and Liqing Zhang. Risk Assessment for Networked-Guarantee Loans Using High-Order Graph Attention Representation. In *IJCAI International Joint Conference on Artificial Intelligence*, 2019.

[3] Wanyu Lin, Zhaolin Gao, and Baochun Li. *Guardian*: Evaluating Trust in Online Social Networks with Graph Convolutional Networks. In *Proc. IEEE International Conference on Computer Communications*, 2020.

[4] Neil Zhenqiang Gong and Bin Liu. You are Who You Know and How You Behave: Attribute Inference Attacks via Users' Social Friends and Behaviors. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 979–995, 2016.

[5] Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. Quantifying Privacy Leakage in Graph Embedding. In *Mobiquitous 2020-17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 1–11, 2020.

[6] Congzheng Song and Vitaly Shmatikov. Overlearning Reveals Sensitive Attributes. In *Prof. International Conference on Learning Representations*, 2020.

[7] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership Inference Attack on Graph Neural Networks. *arXiv preprint arXiv:2101.06570*, 2021.

[8] Gianclaudio Malgieri and Giovanni Comandé. Why a Right to Legibility of Automated Decision-Making Exists in the General Data Protection Regulation. *International Data Privacy Law*, 2017.

[9] Avishek Bose and William Hamilton. Compositional Fairness Constraints for Graph Embeddings. In *International Conference on Machine Learning*, pages 715–724. PMLR, 2019.

[10] Peiyuan Liao, Han Zhao, Keyulu Xu, Tommi Jaakkola, Geoffrey J Gordon, Stefanie Jegelka, and Ruslan Salakhutdinov. Information Obfuscation of Graph Neural Networks. In *International Conference on Machine Learning*, pages 6600–6610. PMLR, 2021.

[11] Thomas N Kipf and Max Welling. Variational Graph Auto-Encoders. In *Proc. NIPS Bayesian Deep Learning Workshop*, 2016.

[12] Ali Makhdoumi, Salman Salamatian, Nadia Fawaz, and Muriel Médard. From the Information Bottleneck to the Privacy Funnel. In *2014 IEEE Information Theory Workshop (ITW 2014)*, pages 501–505. IEEE, 2014.

[13] Borja Rodríguez-Gálvez, Ragnar Thobaben, and Mikael Skoglund. A Variational Approach to Privacy and Fairness. In *2021 IEEE Information Theory Workshop (ITW)*, pages 1–6. IEEE, 2021.

[14] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph Information Bottleneck. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[15] Jing Zhang and Dacheng Tao. Empowering Things with Intelligence: A Survey of The Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2020.

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *Proc. Advances in Neural Information Processing Systems*, 2017.

[17] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in neural information processing systems*, 2016.

[18] Thomas N Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. International Conference on Machine Learning*, 2017.

[19] Petar Veli**v**ković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018.

[20] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph Neural Networks: A Review of Methods and Applications. *AI Open*, 1:57–81, 2020.

[21] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing Links from Graph Neural Networks. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[22] Sina Sajadmanesh and Daniel Gatica-Perez. Locally Private Graph Neural Networks. In *Proc. the ACM Conference on Computer and Communications Security (CCS)*, 2021.

[23] Carl Yang, Haonan Wang, Ke Zhang, Liang Chen, and Lichao Sun. Secure Deep Graph Generation with Link Differential Privacy. *arXiv preprint arXiv:2005.00455*, 2020.

[24] Wanyu Lin, Baochun Li, and Cong Wang. Towards Private Learning on Decentralized Graphs with Local Differential Privacy. *arXiv preprint arXiv:2201.09398*, 2022.

[25] Xiao Han, Yuncong Yang, Leye Wang, and Junjie Wu. Privacy-Preserving Network Embedding against Private Link Inference Attacks. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[26] Binghui Wang, Jiayi Guo, Ang Li, Yiran Chen, and Hai Li. Privacy-Preserving Representation Learning on Graphs: A Mutual Information Perspective. In *Proc. ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1667–1676, 2021.

[27] Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. Invariant Representations without Adversarial Training. In *Advances in Neural Information Processing Systems*, 2018.

[28] Guangxu Mei, Ziyu Guo, Shijun Liu, and Li Pan. SGNN: A Graph Neural Network Based Federated Learning Approach by Hiding Structure. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2560–2568. IEEE, 2019.

[29] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. FedGraphNN: A Federated Learning System and Benchmark for Graph Neural Networks. *arXiv preprint arXiv:2104.07145*, 2021.

[30] Salman Salamatian, Flavio P Calmon, Nadia Fawaz, Ali Makhdoumi, and Muriel Médard. Privacy-Utility Tradeoff and Privacy Funnel. *Unpublished preprint, http://www. mit. edu/ salmansa/files/privacy TIFS. pdf*, 2020.

[31] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: Mutual Information Neural Estimation. In *Proc. International Conference on Machine Learning*, 2018.

[32] Daniel P Palomar and Yonina C Eldar. *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2010.

[33] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S Zemel. The Variational Fair Autoencoder. In *ICLR*, 2016.

[34] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On Variational Bounds of Mutual Information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[35] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating Divergence Functionals and The Likelihood Ratio by Convex Risk Minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[36] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[37] Changsung Moon, Paul Jones, and Nagiza F Samatova. Learning Entity Type Embeddings for Knowledge Graph Completion. In *Proc. ACM Conference on Information and Knowledge Management*, pages 2215–2218, 2017.

[38] F Maxwell Harper and Joseph A Konstan. The Movielens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5(4):1–19, 2015.

[39] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A Semantic Matching Energy Function for Learning with Multi-Relational Data. *Machine Learning*, 94(2):233–259, 2014.

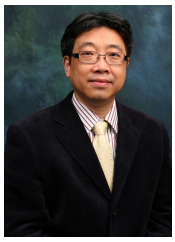
[40] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. International Conference for Learning Representations*, 2015.

[41] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-Based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*, 2019.

[42] Matthias Fey and Jan Eric Lenssen. Fast Graph Representation Learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Wanyu Lin received her Ph.D. degree from the Department of Electrical and Computer Engineering at the University of Toronto. She received her B.Engr. degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology, China and her MPhil. degree from the Department of Computing, The Hong Kong Polytechnic University. Her research interests include graph machine learning, trustworthy machine learning, data privacy, and model interpretability. She has served as associate editor for IEEE Transactions on Neural Networks and Learning Systems (TNNLS). She is a member of IEEE.

Hao Lan received his Ph.D. degree from the Department of Electrical & Computer Engineering at University of Toronto. He recieved both his B.E. and M.E. degree from the School of Communication at Xidian University in 2015 and 2018, respectively. His research interests include distributed machine learning, deep reinforcement learning, graph machine learning, and model interpretability.



Jiannong Cao received the B.Sc. degree in computer science from Nanjing University, China, in 1982, and the M.Sc. and Ph.D. degrees in computer science from Washington State University, USA, in 1986 and 1990, respectively. He is currently the Otto Poon Charitable Foundation Professor in data science and the Chair Professor of distributed and mobile computing with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. He is also the Director of the Internet and Mobile Computing Lab at the Department and the Associate Director of the University Research Facility in big data analytics. His research interests include parallel and distributed computing, wireless networks and mobile computing, big data and cloud computing, pervasive computing, and fault tolerant computing. He has co-authored five books in Mobile Computing and Wireless Sensor Networks, co-edited nine books, and published over 600 papers in major international journals and conference proceedings. He is a Distinguished Member of ACM and a Senior Member of China Computer Federation (CCF).

7 APPENDIX

7.1 Implementation Details

***vGPF* on MovieLens-1M:** MovieLens [38] contains 1,000,209 anonymous ratings from 6,040 MovieLens users on approximately 3900 movies. The main task is to predict the ratings between user and movie with unknown interactions. We model the dataset as an attributed directed graph and the task as link prediction to adopt graph representation learning. In addition, we constraint our method to be privacy-preserving against an adversary attempting to infer user-related attributes from the embeddings. Specifically, the adversary task is to perform node classification for inferring user age, gender, or occupation.

The model architecture for the recommender system on Movielens-1M is illustrated in Fig. 9a. More details are listed below:

- Output embedding dimension: 20
- Number of GNN layers: 2
- Training epochs: 100
- Trained by Adam optimizers [40] with learning rate set to 0.01
- Batch size: 8192
- Train/Val/Test: 0.8/0.1/0.1
- Backbone GNN models: [ChebNet, GCN, GraphSAGE]
- β: [0, 0.005, 0.01, 0.05, 0.1, 0.5]
  - When $\beta = 0$, the sensitive attributes will not be concatenated with the node embeddings $\mathbf{z}$, thus there will be no privacy protection.
  - When $\beta > 0$, the sensitive attributes will be concatenated with the node embeddings $\mathbf{z}$.
- For each experiment we record the test results of the model with the best validation results. And we report the mean and standard deviation of 5 repeated experimental results.
- The attacker model is multi-layer perceptrons with LeakyReLU functions as nonlinearities, taking as input the embeddings output by the best model.
- Attacker training epochs: 100

***vGPF* on FB15k-237 and WN18RR.** Both datasets are knowledge graphs, where nodes represent entities and edges represent relations. In knowledge graphs, complex information is processed into knowledge abstractions represented by simple entity-relation-entity triplets. Predicting relations between entities can help complete a knowledge base. Therefore, the main task on these datasets is link prediction. The adversary tasks on two datasets are listed as follows [9], [10].

- For FB15k-237, the 50-most frequent labels are treated as node-level sensitive attributes [37]. The adversary performs multi-label node classification on 50 labels simultaneously.
- For WN18RR, the word sense and the part-of-speech tag of a node are regarded as two sensitive attributes [39]. The adversary performs multi-class node classification on two attributes separately.

The model architecture for knowledge graphs on FB15k-237 and WN18RR is illustrated in Fig. 9b. More details are listed below:

- Output embeddings dimension: 200
- Number of GNN layers: 1
- Training epochs: 120
- Trained by Adam optimizers with learning rate set to 0.001
- Train/Val/Test: aligns with the one used in the CompGCN paper [41]
- Backbone GNN models: CompGCN
- Batch size: 128
- β (WN18RR): [$1*10^{-9}$, $5*10^{-9}$, $1*10^{-8}$, $5*10^{-8}$,$1*10^{-7}$]
- β (FB15k-237): [10^{-9}, 10^{-8}, 10^{-7}, 10^{-6},10^{-5}]
  (When training *vGPF* on Knowledge Graphs, we set β to smaller values as the loss value is more sensitive to the effects of KL-Divergence term)
- For each experiment we record the test results of the model with the best validation results. And we report the mean and standard deviation of 3 repeated experimental results.
- The attacker model is multi-layer perceptrons with LeakyReLU functions as nonlinearities, taking as input the embeddings output by the best model.
- Attacker training epochs: 30

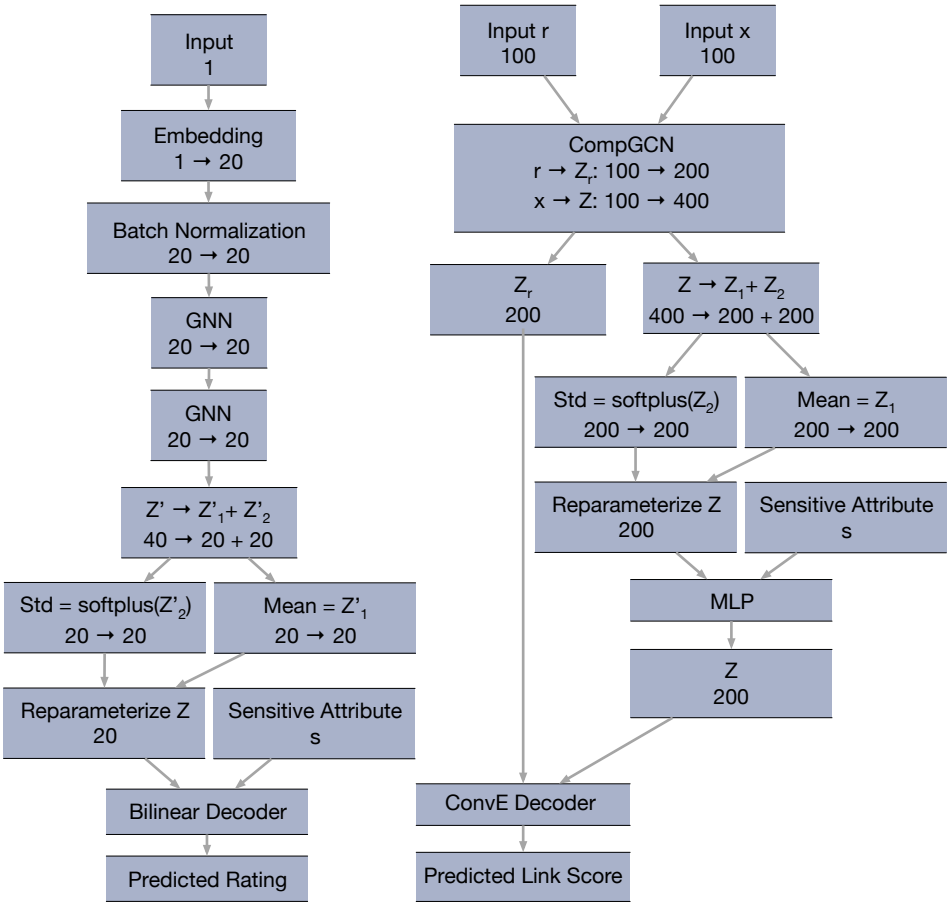
***vGPF* on Citation Networks (Cora, Pubmed, and CiteSeer.** These datasets are citation networks [18], which are widely-used benchmarks in graph learning. Nodes correspond to academic publications and edges are citation links. The main task is link prediction while the adversary attempts to infer the category of certain publications.

The model architecture for citation networks is illustrated in Fig. 10a. More details are listed below:

- Output embedding dimension: 128
- Number of GNN layers: 2
- Training epochs: 150
- Trained by Adam optimizers with learning rate 0.01
- Train/Val/Test: we adopt the default split used in the original paper, maintained by [42]
- Backbone GNN models: [ChebNet, GCN]
- β: [0, 0.0001, 0.001, 0.01, 0.1]
- For each experiment we record the test results of the model with the best validation results. And we report the mean and standard deviation of 5 repeated experimental results.
- The attacker model is a single-layer message-passing module, taking as input the embeddings output by the best model.
- Attacker training epochs: 100

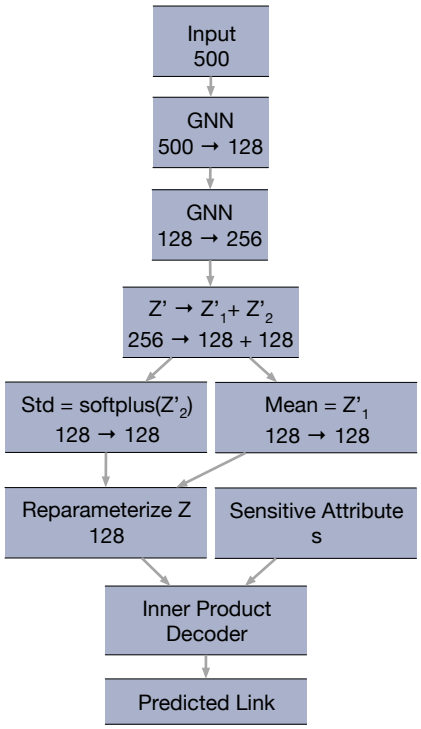
7.2 More Experimental Results

Fig. 11 provides more results for GAL-TV. 11a plot the utility-privacy trade-off of GAL-TV. 11b - 11d illustrate the robustness of GAL-TV against various attribute inference attacks. We can see that GAL-TV and GAL-W exhibit similar performance.



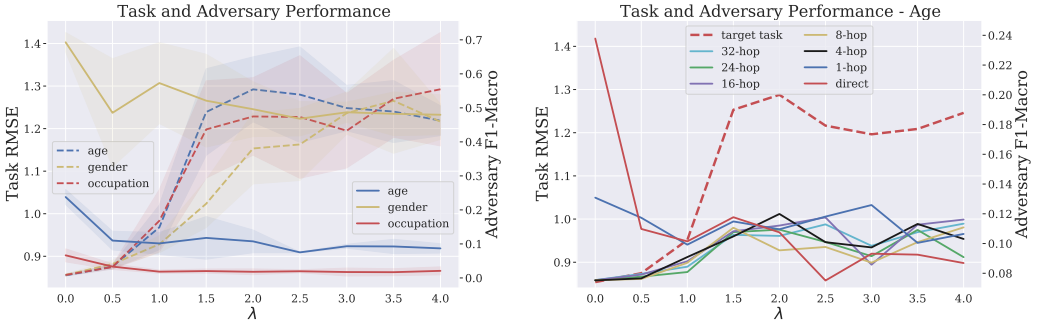
(a) MovieLens-1M (b) FB15k-237 and WN18RR

Fig. 9. Implementations of *vGPF* on MovieLens-1M, FB15k-237, and WN18RR.



(a) Citation Networks

Fig. 10. Implementation Details of *vGPF* on Pumbed. The model architectures for Cora and CiteSeer are the same as Pubmed except for the input dimension.



(a) GAL-TV (b) Age

(c) Gender (d) Occupation

Fig. 11. **More Results for GAL-TV.** 11a plot the utility-privacy trade-off of GAL-TV. 11b - 11d illustrate the robustness of GAL-TV against various attribute inference attacks.